



Adobe® Flash®
Professional CS6

CLASSROOM IN A BOOK®

The official training workbook from Adobe Systems

Instructor Notes

Adobe® Flash® Professional CS6 Classroom in a Book®

© 2012 Adobe Systems Incorporated and its licensors. All rights reserved.

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample files are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, Flash, the Flash logo, Flash Builder, Flash Catalyst, Flash Lite, Flash Player, Classroom in a Book, InDesign, and Photoshop are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apple and Mac OS are trademarks of Apple, registered in the U.S. and other countries. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the U.S. and/or other countries. All other trademarks are the property of their respective owners.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110-2704, USA

Notice to U.S. Government End Users. The Software and Documentation are “Commercial Items,” as that term is defined at 48 C.F.R. §2.101, consisting of “Commercial Computer Software” and “Commercial Computer Software Documentation,” as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Adobe Press books are published by Peachpit, a division of Pearson Education located in Berkeley, California. For the latest on Adobe Press books, go to www.adobe.com. To report errors, please send a note to errata@peachpit.com. For information on getting permission for reprints and excerpts, contact permissions@peachpit.com.

Book:

ISBN-13: 978-0-321-82251-2

ISBN-10: 0-321-82251-X

Instructor Notes:

ISBN-13: 978-0-133-00599-8

ISBN-10: 0-133-00599-2

INSTRUCTOR NOTES

Getting Started

The *Adobe® Flash® Professional CS6 Classroom in a Book®* presents students with a series of real-world projects to learn techniques, solutions, and tips using Adobe Flash Professional CS6 software. The instructor notes are intended to complement the information in the *Adobe® Flash® Professional CS6 Classroom in a Book*. The lessons in this book require assets, including Flash files that are included in the Lessons folder on the book's DVD. Before beginning to work with the lessons, copy the files to the hard drive of each student's computer.

You may want to use all the lessons in this book or just particular lessons. You can use the lessons in any order, but some tools and techniques are described in detail only in the first lesson in which they occur.

Introduce students to additional resources, such as Adobe Help, Adobe TV, Adobe Design Center, and Adobe Developer Connection, where they can access information such as tutorials, updates, and technical help provided by other Flash users and by Adobe.

Course Strategy

This book contains 10 lessons, so if you're teaching a 15-session course, you can spend extra time on some of the more difficult topics and time-consuming lessons. The lessons that deserve at least two class sessions are Lesson 2 on creating graphics, Lesson 4 on animation, Lesson 6 on interactive navigation, and Lesson 8 on sound and video. Depending on the focus of your course, you can spend the remaining extra session on Lesson 5, which covers inverse kinematics and shape tweening; Lesson 7, which covers Text Layout Framework (TLF) text; or Lesson 9, which covers more advanced ActionScript to load external content and masking. If you want to cover new publishing options, such as publishing AIR apps for mobile devices, you can spend more time on Lesson 10.

For beginners, the lessons that should be covered, at an absolute minimum, are Lesson 1, which acquaints the students to the interface and the concept of the Timeline; Lesson 4, which covers motion tweening; and Lesson 6, which deals with interactivity, the button symbol, and ActionScript.

Enhanced and New Features

The enhanced and new features in Flash Professional CS6 are integrated within the lessons of the book.

- Lesson 1 covers the new Document panel, the Autosave and AutoRecovery features, and explains how Flash can automatically resize content when you resize the Stage.
- Lesson 2 details the Convert to Bitmap feature, which is helpful to simplify complex vector art for performance on mobile devices.
- Lesson 3 discusses some of the new options for movie clips, such as the Visible and the Export as Bitmap feature.
- Lesson 4 shows the new integrated Controller panel located at the bottom of the Timeline, as well the ways you can export an animation to a Sprite sheet or PNG sequence.
- Lesson 5 covers the new Pinning feature for inverse kinematics, which helps isolate particular parts of the armature.
- Lesson 6, Lesson 7, and Lesson 8 show readers how to use the revamped Code Snippets panel to simply point to objects on the Stage to add interactivity.
- Lesson 7 covers the new TLF tab ruler and how you can use it to control paragraph indents for more precise layouts.
- Lesson 8 walks readers through the revised Adobe Media Encoder interface.
- Lesson 10 discusses various publishing options, including creating AIR applications for the desktop and AIR apps for mobile devices running Android or the iOS. Readers get a chance to test mobile device interactions in the new Simulator panel. However, a complete project and discussion for mobile devices is beyond the scope of this book because of the required developer certificates and access to the physical device.

Lesson 1: Getting Acquainted

Lesson 1 uses a sample project to provide an overview of the Flash workspace. Students are introduced to the Tools panel, Library panel, Properties inspector, and Timeline. They also learn about layers and keyframes.

Starting Flash and opening a file

Demonstrate two ways of opening a document in Flash: by double-clicking the Flash document or by starting Flash, choosing File > Open. Let your students know that a Flash file can either have the extension .fla or .xfl. The XFL document is a newer Flash format (requiring CS5 or later), which is contained within a folder of

other related assets. Explain the difference between a published SWF file and the FLA or XFL file. A SWF file is the final, published Flash document that is uploaded to a server to play with the Flash Player in a browser, whereas the FLA file is the editable source document.

Emphasize the importance of saving the working copy of the file to the lesson's Start folder. In many lessons, Flash documents reference other asset files, which must be in the same folder as the Flash document.

Getting to know the work area

Identify and describe the following areas of the workspace: Timeline, Tools panel, Stage, Library panel, Properties inspector, and the Edit bar above the Stage. Explain to students how they can undock and move the panels around to suit their working style, and how to return to the default layout by choosing Window > Workspace > Essentials.

If you have Adobe Creative Suite 6 installed, you can show students the similarities between panels in Flash Professional CS6 and panels in other Creative Suite 6 applications. The Timeline is similar to the Timeline in Adobe Premiere Pro and Adobe After Effects. The Properties inspector is similar to the one in Dreamweaver. Many commands work across applications, which will ease the anxiety of learning a new program. For example, copy and paste, or holding down the Shift key to select multiple items, behave the same in Flash.

Understanding the Timeline

Layers overlap each other in the order in which they appear in the Timeline, with the top layer overlapping the bottom layers. As designers plan projects, they should consider the stacking order of the layers and need to be aware of how the objects in different layers appear in front of or behind objects in other layers.

Invite students to move the playhead through (or scrub) the Timeline to see the animation. As you move the playhead, Flash displays the contents of the frame on the Stage. Hide and show different layers on frames to demonstrate how layers are stacked and that each frame may contain content from multiple layers.

Keyframes are essential in animation. A keyframe indicates a change, whether it be in the position or other qualities of something on the Stage or the beginning or end of an audio file. Students will use keyframes in every Flash project, so it's important that they understand the keyframe's purpose clearly.

Describe the difference between inserting a keyframe and inserting a blank keyframe. When you insert a keyframe, the keyframe contains the frame's original content until you change it. When you insert a blank keyframe, all content is removed from the frame so that you can add new content. Also, describe the difference between a frame and a keyframe. A frame is a measure of time on the Timeline. A keyframe is represented on the Timeline with a circle and indicates a change in content on the Stage.

Using the Properties inspector

Demonstrate that the options in the Properties inspector change to reflect what's currently selected. The Properties inspector can display properties for the entire document, the selected frame, the selection on the Stage, or the selected tool in the Tools panel.

Using the Tools panel

The Tools panel contains several tools for selecting, drawing, editing, and navigating. Point out that the main tool for selecting and moving objects is the Selection tool. Demonstrate that some tools are grouped together, and a tool may be hidden beneath another. To select a hidden tool, click the triangle in the icon of the displayed tool, and then select the hidden tool from the pop-up menu.

Beneath the tools is the tool's options area. The options available depend on the selected tool. Demonstrate how the options area changes as you select different tools. When you select the Rectangle tool, the Object Drawing mode icon appears; when you select the Zoom tool, the Enlarge and Reduce icons are available; when you select the Pencil tool, the Pencil mode icons are available. Show students that additional options for a selected tool are also available in the Properties inspector.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

- 1 What is the difference between the FLA file, the XFL file, and the SWF file, and how do you create them?
- 2 What do you do when you can't find a panel that you know exists?
- 3 How does Flash measure its X and Y coordinates on the Stage?
- 4 How do you change the dimensions of the Stage and keep the contents proportional?
- 5 If your panels have been moved out of their normal positions, how do you put them back to their default positions?
- 6 What is the difference between the AutoSave and AutoRecovery features?

Answers

- 1 The FLA and the XFL are both editable source files for a Flash project. When you save a Flash project, you can save it as a FLA document or as the newer uncompressed XFL document, which exposes the contents of the Flash file for other developers. The SWF is the exported file that is generated when you publish a movie for the Flash Player or test a movie (Control > Test Movie > in Flash Professional). The SWF is the file that is uploaded to your Web server to play in a browser.
- 2 You can access all the panels by choosing Window from the top menu.
- 3 The X and Y values are measured on the Stage from the top-left corner. X begins at 0 and increases to the right, and Y begins at 0 and increases downward.
- 4 In the Properties inspector, click the Edit button next to Size. In the Document Settings dialog box that appears, enter new pixel values in the Dimensions fields and choose “Scale content with stage.”
- 5 Choose Window > Workspace > Reset Essentials to reset the Essentials workspace, which is the default arrangement of panels.
- 6 The AutoSave feature saves your open working document at regular, specified intervals. The AutoRecovery feature saves a copy of your document in case of a crash, and the copy is deleted once you successfully close your document or quit the Flash application.

Lesson 2: Working with Graphics

Lesson 2 uses a sample project to introduce students to the drawing tools and to creating simple graphics. The project is a static banner ad showing a mug of steaming coffee over a wavy background with floral decorations and a text logo. As they create the banner, students learn how to use the various drawing, editing, and selection tools in the Tools panel. They learn to create basic shapes, import a bit-map fill, make selections, group objects, make patterns and decorations and align objects, create gradients and transparencies, create curves, and manipulate objects with the Free Transform tool. They also learn the difference between drawing modes in Flash. Be sure to introduce the new Convert to Bitmap feature, which can be helpful to simplify artwork and reduce the performance load on mobile devices.

Flash drawing modes

By default, Flash uses the merge drawing mode when you draw rectangles, ovals, or lines. In merge drawing mode, the drawn elements are called shapes. Overlapping shapes merge together to form a single element on the Stage. When you select a shape, the selected area appears dotted. Though the elements merge together, you

can select each separately. Each side of a rectangle's stroke is a separate element, for example. To select a shape's fill and stroke, drag the Selection tool around the entire shape. Demonstrate to students how to select the fill or the stroke of a shape by clicking on it, or to select the entire shape (the fill and the stroke) by double-clicking on it.

In object drawing mode—active when the Object Drawing icon is selected in the options area of the Tools panel—rectangles, ovals, and lines are discrete elements, even when they overlap. The discrete elements are called drawing objects. When you select a drawing object, its fill and stroke are both selected.

In primitive drawing mode—active when you use the Rectangle Primitive or Oval Primitive drawing tools—drawn elements are discrete objects, but you can modify their corner angles or inner radius.

Merge drawing mode offers advantages in creating unusual, complex shapes.

Object drawing mode lets you move objects without concern that they will overlap and clip other objects. Primitive drawing mode makes it possible to draw more complex shapes without having to merge multiple shapes together.

Invite students to draw and manipulate elements using each of these drawing modes so they become familiar with the differences and can recognize them as they work with them. You can convert a shape to an object by selecting it and choosing **Modify > Combine Object > Union**. You can convert an object to a shape by selecting it and choosing **Modify > Break Apart**.

Breaking apart and grouping objects

Emphasize for students the difference between shapes and groups in Flash, and the advantage of groups when you want to align certain elements of a drawing with the Align panel. Demonstrate how you can have groups within groups, and mention that you can always break apart groups to their component parts.

Creating transparencies

Transparency, or inversely, opacity, is determined by the alpha value of a color. Decreasing the alpha value decreases the opacity and increases the transparency. Alpha values range from 0 (totally transparent) to 100 (totally opaque). Students can change the alpha value for a color in the Color panel or in the Tools panel, under Fill Color options, in the Alpha field.

Creating gradients

In a gradient, one color gradually changes into another. Flash can create linear gradients, which change color horizontally, vertically, or diagonally; or radial gradients, which change color moving outward from a central focal point. Encourage students to apply a radial gradient and a linear gradient in succession to the same

shape to see how the gradients differ. Students can add color pointers beneath the gradient definition bar in the Color panel to add up to 15 color transitions. In addition to choosing colors and positioning the color pointers for a gradient, demonstrate how you can adjust the size, direction, or center of a gradient fill with the Gradient Transform tool.

Creating and editing text

Creating text in Flash is similar to creating text in Adobe InDesign, Adobe Photoshop, or numerous other applications. Use the Text tool to create an insertion point on the Stage and begin typing, or drag the Text tool to create a text box for your text. In this lesson, students use Classic text, which is used for text that doesn't use the new Text Layout Framework feature, which is covered in more detail in Lesson 7. To format text, select it with the Text tool, and then select formatting options in the Properties inspector. To manipulate a text box as an object, select it with the Selection tool and make changes in the Properties inspector or Transform panel, or simply drag the text box handles to resize it. Resizing the text box does not resize the text, but it does reflow the text in the text box.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

- 1 How do you create a corner point when editing the contours of a shape with the Selection tool?
- 2 What are two ways you can change the fill of a shape?
- 3 What's the difference between a linear gradient and a radial gradient?
- 4 What is the function of the Lock Fill option for the Paint Bucket tool?

Answers

- 1 Hold down the Alt (Windows) or Option (Mac) key while dragging the sides of a shape to add a new corner.
- 2 One way you can change the fill of a shape is to choose the Paint Bucket tool, choose a new Fill color in the Properties inspector, and then click on the shape. The other way is to select the fill on the Stage, and then choose a new color in the Properties inspector.
- 3 A linear gradient changes color horizontally, vertically, or diagonally, whereas a radial gradient changes color moving outward from a central focal point.
- 4 The Lock Fill option locks the current gradient to the first shape where it was applied so that subsequent shapes extend the gradient.

Lesson 3: Creating and Editing Symbols

Lesson 3 uses a sample project to introduce symbols, instances, and imported Illustrator and Photoshop files as students create a static frame of a cartoon. Students learn to create new symbols and to convert selections on the Stage to symbols. They learn how to edit the symbols from the Library panel and to edit the symbols in place on the Stage. Students also learn to adjust the color effect and blending, apply filters, and move the symbol instances in three-dimensional space.

Importing Illustrator files

Flash lets you import Illustrator AI files and to a large extent preserves the artwork's editability and visual fidelity. The AI Importer also provides a great degree of control in determining how Illustrator artwork is imported into Flash, letting you specify how to import specific objects into an AI file.

The Flash AI Importer provides the following key features:

- Preserves editability of the most commonly used Illustrator effects such as Flash filters.
- Preserves editability of blend modes that Flash and Illustrator have in common.
- Preserves the fidelity and editability of gradient fills.
- Maintains the appearance of RGB (red, green, blue) colors.
- Imports Illustrator Symbols as Flash Symbols.
- Preserves the number and position of Bezier control points.
- Preserves the fidelity of clip masks.
- Preserves the fidelity of pattern strokes and fills.
- Preserves object transparency.
- Converts the AI file layers to individual Flash layers, keyframes, or a single Flash layer. You can also import the AI file as a single bitmap image, in which case Flash flattens (rasterizes) the file.
- Provides an improved copy-and-paste workflow between Illustrator and Flash. A copy-and-paste dialog box provides settings to apply to AI files being pasted onto the Flash Stage.

You specify how artwork is imported in the Import dialog box. You can select which layers—and which content on each layer—to import. Flash can import Illustrator layers as Flash layers, as a single layer, as keyframes, or as symbols.

In this lesson, students import an Illustrator file with the characters of the cartoon frame. Many of the same options are available when copying and pasting artwork. If Illustrator is installed on the computer, open the Illustrator file and copy and paste, or drag and drop, artwork from it to the Stage in Flash to demonstrate how the options are similar.

About the FXG file format

The FXG file format is a cross-platform graphics file format that you can also use to easily move your artwork between Flash and other Adobe graphics programs such as Illustrator. If you want to export your Flash artwork as an FXG file, choose File > Export > Export Image and select Adobe FXG. Import FXG artwork as you would any other external file by choosing File > Import To Stage, or File > Import To Library.

About symbols

A *symbol* is a reusable asset that you can use for special effects, animation, or interactivity. There are three kinds of symbols: the graphic, button, and movie clip. Symbols can reduce the file size and download time for many animations because they can be reused. You can use a symbol countless times in a project, but Flash includes its data only once.

Symbols are stored in the Library panel. When you drag a symbol to the Stage, Flash creates an *instance* of the symbol, leaving the original in the library. An instance is a copy of a symbol located on the Stage. You can offer your students the analogy of a symbol as a photographic negative. The instances on the Stage can be thought of as prints of the negative. With just a single negative, you can create multiple prints. It's also useful to talk about symbols as containers. Like containers, symbols hold different kinds of content—whether it's an imported Illustrator graphic, a bitmap, or a piece of text. You can always edit or replace the contents of the symbol, just like you can edit or replace the contents of a container.

Movie clip symbols, button symbols, and graphic symbols each have unique functions, benefits, and limitations. Create one of each, and then show students the differences in the Timeline and in the Properties inspector for each.

Creating symbols

In Flash, there are two ways to create a symbol. The first is to have nothing on the Stage selected, and then choose Insert > New Symbol. Flash brings you to symbol-editing mode, where you can begin drawing or importing graphics for your symbol.

The second way is to select existing graphics on the Stage, and then choose Modify > Convert to Symbol (or press F8). Whatever is selected will automatically be placed inside your new symbol. The “Convert to Symbol” language may confuse students, because there is really no “conversion.” Explain that the selected items are simply placed inside the newly created symbol.

The registration point of a symbol is indicated by the crosshair on the Stage in symbol-editing mode. The registration point determines the point at which Flash reports the coordinates of the symbol and from which it resizes or transforms the symbol.

Editing and managing symbols

Demonstrate to students the two ways in which you can edit your symbols. First, you can double-click the symbol in the Library panel and edit its contents in symbol-editing mode. Return to the main Timeline by clicking the Scene 1 button on the top horizontal bar above the Stage. Second, you can double-click the instance on the Stage and edit the symbol in place. Editing in place lets you see all the other elements surrounding the instance on the Stage. When you edit a symbol, all of its instances on the Stage change to reflect the new edits.

Changing the appearance of instances

You can change the appearance of instances in a number of ways. You can change the size of any instance with the Free Transform tool. You can also change the brightness, tint, or transparency of an instance by setting different values for the instance's Color Effect in the Properties inspector. You can also apply a Blend effect or a Filter on an instance. Blending refers to how the colors of an instance interact with the colors below it. Invite students to explore the different Blend options. Filters are special effects, such as blurs and glows, which you can apply to instances. Additionally, you can position or rotate any instance in three-dimensional space using the 3D Translation and 3D Rotation tool.

The new Visible and Render options in the Properties inspector can help you manage movie clip instances. The Visible option is an easy way to position an instance on the Stage and control its visibility at a later point in time, either with ActionScript or with the Visible option. The Render options can help performance on mobile devices by simplifying the display of complicated vector art as bitmap art.

Using rulers, guides, and grids

Explain to students that many tools can help them position instances and other objects on the Stage. In Lesson 1, students used the Properties inspector to position objects by changing the X and Y coordinates, and in Lesson 2, they learned to use the Align panel to align several objects to each other. Another way to position objects on the Stage is to use rulers, guides, and grids. Rulers (View > Rulers) appear on the top and left edge of the Pasteboard to provide measurement along the horizontal and vertical axes. Guides (View > Guides > Show Guides) are vertical or horizontal lines that you can pull out from the rulers and position on the Stage but that do not appear in the final published movie. The grid (View > Grid > Show Grid) superimposes vertical and horizontal lines on the Stage and also does not appear in the final published movie. Another useful feature is the Move Guide option, which appears when you double-click on a guide. In the Move Guide dialog box, you can enter pixel values to position a guide exactly where you want it.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

- 1 What are the three kinds of symbols, and how do they differ?
- 2 When you import a Photoshop document into Flash, how do you preserve the layer effects?
- 3 How do you convert a bitmap image into a vector graphic?
- 4 What are blending effects?
- 5 What is the difference between the Global and Local Transform option for the 3D Translation and 3D Rotation tool?
- 6 How do you reset the transformations of an instance on the Stage?

Answers

- 1 The three symbols are Graphic, Button, and Movie Clip. A Graphic symbol is the most basic of symbols, which you can use for animation, but does not support ActionScript, blending effects, or filters. A Button symbol is used for interactivity and contains four special keyframes that describe how it appears when the mouse is interacting with it. A Movie Clip symbol contains its own independent Timeline. You can apply filters, color settings, and blending modes to a Movie Clip instance, and you can use ActionScript to control a Movie Clip instance.
- 2 In the Import to Stage dialog box, choose the "Bitmap image with editable layer styles" option. The Movie Clip symbol icon appears to the right of the Photoshop layer, indicating that the imported layer will be made into a Movie Clip symbol. Any layer effects such as transparencies or blending will be preserved.
- 3 To convert a bitmap to a vector, import the bitmap image into Flash. Select the bitmap and choose Modify > Bitmap > Trace Bitmap. The options in the dialog box that appears determine how faithful of a trace the vector image will be to the original bitmap.
- 4 Blending refers to how the colors of an instance interact with the colors below it.
- 5 When the Global Transform option is selected, rotation and positioning is relative to the global, or Stage, coordinate system. The 3D display over the object that you're moving shows the three axes in constant position, no matter how the object is rotated or moved. When the Global Transform option is not selected, rotation and positioning is relative to the object. The 3D display shows the three axes oriented relative to the object, not the Stage.
- 6 If you want to reset the transformations of your instance, you can use the Transform panel (Window > Transform). Click the Remove Transform button in the lower-right corner of the Transform panel to reset the instance to its original settings.

Lesson 4: Adding Animation

Lesson 4 uses an animated motion picture promotional Web site to introduce students to motion tweens, easing, editing the paths of motion, the Motion Editor, and nested animations.

About animation

Animation is the movement, or change, of objects through time. Animation can be as simple as moving a box across the Stage from one frame to the next. It can also be much more complex, involving properties other than an object's location. You can animate many aspects of a single object: the position, the color or transparency, the size or rotation, and filters. You can also animate an object's position and rotation in three-dimensional space. You can control its path of motion, and its easing, which is the way an object accelerates or decelerates.

The basic workflow for creating motion goes like this: To animate objects in Flash, you select the object on the Stage, right-click/Ctrl-click, and choose Create Motion Tween from the context menu. Move the playhead to a different point in time and move the object to a new position.

Motion tweens require you to use a symbol instance. If the object you've selected is not a symbol instance, Flash will automatically ask to convert the selection to a symbol. Flash also automatically separates motion tweens on their own layers, which are called Tween layers. There can only be one motion tween per layer without any other element in the layer. Tween layers allow you to change various attributes of your instance at different key points over time. It's helpful to think of Motion tweening as "instance tweening," since you are changing the properties of an instance over time. This will help students differentiate between Motion tweening and Shape tweening, which they will encounter in the next lesson.

Changing the pacing and timing

Simply click and drag the end of a tween span to add more frames to it and expand the amount of time it takes to do the animation. All the keyframes within the tween span will be distributed evenly. You can also Shift-drag the end of the tween span to add frames without affecting the timing of the keyframes. Remind students that they can also right-click/Ctrl-click on the Timeline and choose Insert Frames to add frames, or press F5 on the keyboard to add frames.

To move individual keyframes, first Ctrl-click (Windows) or Command-click (Mac) to select the keyframe, and then click and drag it to a new position.

Demonstrate to students how the positions of the keyframes affect the timing of the animation and how the total length of the tween span affects the overall speed of the animation.

Changing the path of the motion

The path that an object takes as it moves on the Stage during a motion tween is the motion path, and it is represented by a colored line. You can edit the path of the motion with the Free Transform tool, or by using the Selection tool, or, if you want Bezier precision, you can use the Convert Anchor Point tool and Subselection tool.

Creating nested animations

Movie clip symbols have their own Timelines, similar to the main Timeline. A movie clip symbol may have other symbols nested within it. In this lesson, students create a nested animation of the alien waving his arms and another nested animation of the car jerking up and down. Ask your students to brainstorm what other kind of nested animations would require an animation inside a movie clip symbol.

For nested animations, demonstrate the new export functions that generate a PNG sequence or a sprite sheet. PNG sequences and sprite sheets are used as animation assets for alternative (non-Flash) ways to create animation.

Using the Motion Editor

The Motion Editor is a panel that provides in-depth information and editing capabilities for all the properties of a motion tween. The Motion Editor is located behind the Timeline and can be accessed by clicking on the top tab or by choosing Window > Motion Editor.

The Motion Editor shows how individual properties change over time as a graph. You can add keyframes to a graph by clicking on the diamond icon or right-clicking/Ctrl-clicking on the graph and choosing Insert Keyframe. Drag the keyframe up or down to its new value, or enter the new amount in the Value column.

You can use the Motion Editor to apply sophisticated easing effects to any property. Easing refers to the way in which a motion tween proceeds. In the most basic sense, it can be thought of as acceleration or deceleration. Right-click/Ctrl-click a keyframe in the Motion Editor and select Smooth point to make handles appear on the keyframe. Drag the handles to change the curvature of the graph, which affects the easing for that property. Envisioning easing as a graph may be difficult for students, so describe many different easing situations and present their corresponding graph.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

- 1 Why is animation referred to as “tweening”?
- 2 If you want to animate an airplane moving across a static background, how many layers are required?
- 3 Identify two ways that you can lengthen the duration of a motion tween.
- 4 What is the Motion Presets panel and how do you use it?
- 5 What does the “Orient to path” option do?
- 6 What’s the difference between an ease-in and an ease-out?

Answers

- 1 The term “tweening” comes from the world of classic animation. Senior animators would be responsible for drawing the beginning and ending poses for their characters. The beginning and ending poses were the keyframes of the animation. Junior animators would then come in and draw the “in-between” frames, or do the “in-betweening.” Hence, “tweening” refers to the smooth transitions between keyframes.
- 2 You need two layers: one layer to create a motion tween for the plane and a second layer for the static background.
- 3 To lengthen the duration of a motion tween, you can drag the end of a tween span to lengthen its total time. You can also choose Insert > Timeline > Frame (or press F5) to add frames to a tween span.
- 4 The Motion Presets panel (Window > Motion Presets) stores particular motion tweens so you can apply them to different instances on the Stage. To save a motion tween, select the motion tween on the Timeline or the instance on the Stage. In the Motion Presets panel, click the Save selection as preset button and name your motion tween to be stored in the Motion Presets panel.
- 5 The “Orient to path” option in the Properties inspector adjusts the object traveling along its path so it is oriented in the direction in which it is heading.
- 6 Easing refers to the way in which a motion tween proceeds. An ease-in refers to motion that gradually begins from its starting keyframe. An ease-out refers to motion that gradually slows down as it approaches its ending keyframe.

Lesson 5: Articulated Motion and Morphing

Lesson 5 introduces methods for animating two specialized kinds of motion: articulated motion and morphing. Articulated motion involves motion of an object that is made up of multiple, linked objects. Flash can animate a linked object (an *armature*) using inverse kinematics, a mathematical way to calculate how to move an armature from one pose to another. Morphing is the animation of shapes and can create more organic changes in contours. Students explore both kinds of specialized animations with an animated crane, a floating buoy, and the ocean in one file, and an octopus in another file.

Articulated motion with inverse kinematics

The first step to create articulated motion is to define the bones of an object. You use the Bone tool to do that. The Bone tool tells Flash how a series of movie clip instances are connected. The connected movie clips are called the *armature*, and each movie clip is called a *node*. Armatures are separated on a Pose layer. Students learn to pose the crane at one point in time, and then create a new pose at a later point in time, and watch how Flash automatically animates the motion from one pose to the next.

Demonstrate how you can isolate the movement of each node by holding down the Shift key as you move it. You can also edit the positions of the nodes by holding down the Alt (Windows) or Option (Mac) key and dragging them to new positions.

Constraining joints

Students are asked to constrain the various joints of the crane to explore the possibilities of the option. Many armatures in real life are constrained to certain angles of rotation. For example, your forearm can rotate up toward your bicep, but it can't rotate in the other direction beyond your bicep. When working with armatures in Flash, you can choose to constrain the rotation for various joints or even constrain the translation (movement) of the various joints. Click on a bone in an armature and select Constrain in the Properties inspector. Set the minimum and maximum allowable angles to constrain the joint.

The new Pinning feature for inverse kinematics can help pose your armature. You can select the Pin option in the Properties inspector, or click on a specific joint on the Stage to fix the armature at that point, leaving the bones further down the hierarchy free to move.

Inverse kinematics with shapes

You can also create armatures for shapes, which are useful for animating objects without obvious joints and segments but can still have an articulated motion. In this part of the lesson, students add bones to the arms of an octopus to animate its undulating motion. To create an armature for a shape, select the Bone tool. Click and drag inside the selected shape to define the first bone and the subsequent connected bones.

The Bind tool can refine the connections between the shape and its armature. The Bind tool is hidden under the Bone tool. The Bind tool displays which control points are connected to which bones and lets you break those connections and make new ones.

Choose the Bind tool and click on any bone in the shape. The selected bone is highlighted in red, and all the connected control points on the shape are highlighted in yellow. If you want to redefine which control points are connected to the selected bone, you can do the following:

- Shift-click to add additional control points.
- Ctrl-click/Command-click to remove control points.
- Drag a connection line between the bone and the control point.

Click on any control point on the shape. The selected control point is highlighted in red, and all the connected bones are highlighted in yellow. If you want to redefine which bones are connected to the selected control point, you can do the following:

- Shift-click to add additional bones.
- Ctrl-click/Command-click to remove bones.
- Drag a connection line between the control point and the bone.

Morphing with shape tweens

Shape tweening is a technique for interpolating amorphous changes between shapes in different keyframes. Shape tweens make it possible to smoothly morph one thing into another. Any kind of animation that requires that the contours of a shape change—for example, animation of clouds, water, or fire—is a perfect candidate for shape tweening. Explore with your students different animation challenges and have them identify which techniques would be most appropriate to complete the animation. In this part of the lesson, students create two keyframes, each containing different shapes of the surface of the ocean. Applying a shape tween in between the two keyframes creates a smooth animation that morphs the shapes.

Emphasize to students that both the fill and the stroke of a shape can be smoothly animated. Also, point out that because shape tweening only applies to shapes, you can't use groups, symbol instances, or bitmap images.

Using shape hints

Shape hints force Flash to map points on the first shape to corresponding points on the second shape. By placing multiple shape hints, you can control more precisely how a shape tween appears. Insert a shape hint by selecting the first keyframe of a shape tween. Then choose **Modify > Shape > Add Shape Hint**. A tiny, circled letter appears in the first keyframe and a corresponding circled letter appears in the next keyframe. Move both circled letters to matching points on the contours of the shapes.

A maximum of 26 shape hints are allowable; for best results, place shape hints in a clockwise or counterclockwise fashion.

Using the spring option to simulate physics

The spring option can be applied to any bone of an armature to determine how rigid or flexible it behaves as the whole armature moves. For example, a flexible object (like a flag or a flag pole) normally would have some “springiness” that would cause it to jiggle on its own as it moves and continue to jiggle even after motion of the entire body stops. Ask students to think where they could apply the spring option to make a more realistic animation. Remind students that the spring option can be applied only to animations using inverse kinematics. The spring option can be applied for armatures that use movie clips or for armatures in shapes.

Questions

The following are additional questions that are not in the students’ Classroom in a Book.

- 1 What does parent, child, and sibling refer to in an armature?
- 2 How do you isolate the rotation of individual nodes in an armature?
- 3 What is a runtime armature, and what is its major restriction?
- 4 How do you remove shape hints?

Answers

- 1 The first bone of an armature is referred to as the parent, and the bone that is linked to it is called the child. A bone can have more than one child attached to it as well. For example, an armature of a puppet would have a pelvis connected to two thighs, which in turn are attached to two lower legs of their own. The pelvis is the parent, each thigh is a child, and the thighs are siblings to each other.
- 2 Hold down the Shift key to isolate the rotations of individual nodes so that you can position your poses exactly as you want them. You can also use the Pin option to fix certain joints to the Stage.

- 3 A runtime armature is interactive and allows the user to move the armature. You can make any of your armatures—whether they are made with a series of movie clips or made with a shape—into a runtime armature. Runtime armatures, however, are restricted to armatures that only have a single pose.
- 4 To remove a shape hint, drag it entirely off the Stage and Pasteboard. To remove all shape hints, choose **Modify > Shape > Remove All Hints**. Removing a shape hint in one keyframe will remove its corresponding shape hint in the other keyframe.

Lesson 6: Creating Interactive Navigation

Lesson 6 introduces ActionScript and the button symbol. Students learn how to structure the Timeline to create a project with nonlinear navigation—a project in which the user chooses what content to see and when to see it. Students learn about basic event handling and the commands for controlling the playhead. The end result is an interactive restaurant guide in which the user can click on any thumbnail image of a restaurant to see more information and reviews of that restaurant.

About interactive movies

Students should learn what it means to create nonlinear navigation. In this lesson, students create a Flash movie that doesn't play straight from the beginning to the end. ActionScript provides the instructions to make the Flash playhead to jump around and to go to different frames of the Timeline based on which button the user clicks. Different frames on the Timeline contain different content. Users don't actually know that the playhead is jumping around the Timeline—all they see (or hear) is different content appearing as they click the buttons on the Stage.

Creating buttons

The buttons in this lesson include small thumbnail bitmap images with some text. Review the three kinds of symbols and the benefits of each. Button symbols contain four unique keyframes. Demonstrate and explain each keyframe: Up, Over, Down, and Hit:

- The Up state determines the button's default appearance when the mouse is not interacting with it.
- The Over state appears when the mouse rolls over the button.
- The Down state appears when the mouse is pressed over the button.
- The Hit state defines the button's active area, which is the area that responds to the mouse rolling over or clicking the button.

A particularly effective demonstration of the Hit state is to delete half of the Hit state graphic in the button symbol. Test the movie with the button instance on the Stage and show how only the remaining half of the button is responsive to the mouse.

For more advanced students, show how button symbols with only their Hit states defined can be used as invisible buttons. Place an invisible button instance on the Stage to make that particular area clickable and responsive to the mouse.

Naming the button instances

Giving names to instances is a critical step in creating interactive Flash projects. Emphasize the importance of naming instances so ActionScript can refer to them. The most common beginner mistake is not to name, or to incorrectly name, a button instance. The instance names are not the same as the symbol names in the Library panel. The names in the Library panel are simple organizational reminders.

Instance naming follows these simple rules:

- Do not use spaces or special punctuation. Underscores are OK to use.
- Do not begin the name with a number.
- Do not use any words that are also ActionScript keywords or commands.
- Be mindful of uppercase and lowercase letters.

One source of confusion is the optional, but helpful, `_btn` suffix. Some students think that the suffix is only required in the Instance Name field in the Properties inspector but not in the actual ActionScript coding, or vice versa. Remind them that the names must match exactly.

Name instances in a way that is meaningful, so you'll remember what you've called them when you're writing ActionScript. Shorter names are easier to type without error, but longer names may be easier to remember and to identify when you're editing the ActionScript later.

Understanding ActionScript 3.0

Adobe Flash Professional CS6 uses ActionScript 3.0, the scripting language that extends Flash functionality and enables interactivity in Flash projects. Students who haven't worked with scripting languages before may initially be intimidated and confused by ActionScript, which is an object-oriented scripting language. This lesson introduces basic scripting concepts, terminology, and syntax, as well as specific ActionScript coding to make their buttons functional.

About ActionScript

The ActionScript scripting language lets you add complex interactivity, playback control, and data display to your application. You can add ActionScript in the authoring environment by using the Actions panel.

ActionScript follows its own rules of syntax, includes reserved keywords, and lets you use variables to store and retrieve information. ActionScript includes a large library of built-in classes that let you create objects to perform many useful tasks. For detailed information on ActionScript, tell students that they can always refer to *Programming ActionScript 3.0* or the *ActionScript Language References*; both are available in Flash Help. It's particularly important for students to be able to find and learn how to use an ActionScript element on their own by reading the Help entries.

You don't need to understand every ActionScript element to begin scripting; if you have a clear goal, you can start building scripts with simple actions.

Because there are multiple ways of incorporating ActionScript into your Flash document files, there are several different ways to learn ActionScript.

Flash Help describes the graphical user interface for working with ActionScript. This interface includes the Actions panel, Script window, Script Assist mode, Code Snippets panel, Output panel, and Compiler Errors panel.

Dealing with errors

All students struggle with syntax errors when they learn ActionScript. Explain to your students that this is a normal process, and even veteran developers deal with debugging. Share with them the ways in which they can identify and fix the mistakes in their code:

- Use the Check syntax button above the Script pane in the Actions panel to have Flash check their code for syntax errors.
- Use the Auto format button above the Script pane in the Actions panel to check for errors and reformat their code to make it easier to read.
- Use the errors that appear in the Compiler Errors panel to identify the kind and source of the problem.
- Use the color-coding of the code as it appears in the Script pane to check if the correct keywords are colored.
- More advanced students can begin to use the `trace()` command to send messages to the Output panel.

Adding a stop action

Students have had some experience with ActionScript and the Actions panel in earlier lessons because they've added stop actions. Stop actions are very simple; they stop the movement in the Timeline and prevent files from looping.

As you discuss stop actions, talk about the role of stop actions added in earlier lessons. Remove a stop action from the end file of an earlier lesson and preview the movie. Discuss what happens when the stop file is no longer there. In most cases, the movie or some part of the movie repeats.

Discourage the use of Script Assist to add ActionScript to the Actions panel. Although it is meant to help users unfamiliar with coding, it is far more useful to teach students how to write code in the Script pane unassisted rather than hunting for the code element in the Actions Toolbox and filling out the supplied fields. Students will have better success at automatically adding ActionScript with the Code Snippets panel, discussed later in the lesson.

Creating event handlers for buttons

Events happen in the Flash environment that Flash can detect and respond to. Explain the concept of an event to students, and give examples of various types of events. For example, a mouse click, a mouse movement, and a key press on the keyboard are all events. Other mobile device interactions are also events, such as pinching, swiping, or even accessing the accelerometer by tilting the device. These events are produced by the user, but some events can happen independently of the user, like the successful loading of a piece of data or the completion of a sound. With ActionScript, you can write code that detects events and responds to them with an event handler.

The first step in event handling is to create a listener that will detect the event. A listener looks something like this:

```
wheretolisten.addEventListener(whatevent, responsetoevent);
```

The actual command is `addEventListener()`. Clarify to students which of the words are placeholders for objects and parameters for their situation. `Wheretolisten` is the object where the event occurs (usually a button), `whatevent` is the specific kind of event (such as a mouse click), and `responsetoevent` is a function that gets triggered when the event happens.

The next step is to create the function that will respond to the event. A function simply groups a bunch of actions together, which you can trigger by referencing its name. A function looks something like this:

```
function responsetoevent (myEvent:MouseEvent){ };
```

In this particular example, the name of the function is called `responsetoevent`. It receives one parameter (within the parentheses) called `myEvent`, which is an event

that involves a mouse event. If this function is triggered, all the actions between the curly braces are executed.

Students should memorize the most common mouse events and ActionScript commands for navigating the Timeline. The mouse events they should remember are

- `MouseEvent.CLICK`
- `MouseEvent.MOUSE_MOVE`
- `MouseEvent.MOUSE_DOWN`
- `MouseEvent.MOUSE_UP`
- `MouseEvent.MOUSE_OVER`
- `MouseEvent.MOUSE_OUT`

The commands to control the playhead that they should remember are

- `stop();`
- `play();`
- `gotoAndStop(framenumber or "frameLabel");`
- `gotoAndPlay(framenumber or "frameLabel");`
- `nextFrame();`
- `prevFrame();`

Creating destination keyframes

When the user clicks each button, Flash will move the playhead to a new spot on the Timeline. Students are asked to create new content at those particular frames. In this lesson, students learn to put new content in different keyframes and to label each keyframe to make referencing each from ActionScript easier.

A few essential points for students to remember: Use double quotes around the name of the frame label in their ActionScript code. If they want to use the `gotoAndPlay()` command, they will also need to add a `stop()` command on the Timeline to stop the playhead before it displays all the other keyframes along the Timeline.

Using the Code Snippets panel

The Code Snippets panel provides commonly used ActionScript code for you to easily add simple interactivity to your Flash project. If students are unsure of coding their own buttons, they can use the Code Snippets panel to learn how to add interactivity. Flash CS6 introduces a new way of adding code to objects on the Stage by simply pointing from the code to the object with the pick whip. You can also save, import, and share code between a team of developers to make the development and production process more efficient. In this lesson, students use the Code Snippets panel to add interactivity to their home button.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

- 1 What happens if a button symbol has an empty keyframe in its Up state but filled keyframes in its Over, Down, and Hit states?
- 2 What is the significance of the color of a shape in a button symbol's Hit keyframe?
- 3 In what kind of situation would you want a button's Hit state to be larger than its Up state?
- 4 What is wrong with the instance name of this button, 02restaurantbtn, and how would you fix it?
- 5 What additional information is needed in between the parentheses of the command `gotoAndStop()`? What information is needed in between the parentheses of the command `stop()`?
- 6 How do you use the Code Snippets panel to program a button to go to a specific frame?

Answers

- 1 If a button symbol has an empty keyframe in its Up state, the button instance on the Stage is invisible. The mouse still can interact with the button instance, but the user won't be able to see it.
- 2 There is no significance to the color of a shape in a button symbol's Hit keyframe. Only the size and position of a shape in the Hit state determines the clickable area.
- 3 You may want the clickable area to cover a larger area than the visible button. For example, a typical pause button on consumer electronics is represented by two parallel vertical lines. You would want the Hit state to cover the gap between the vertical lines so users have an easier time clicking the button.
- 4 The instance name cannot begin with a number. You may want to add an underscore before `btn`, but it is not incorrect without it.
- 5 The `gotoAndStop()` command requires the destination frame number or name of the frame label, in quotation marks. The `stop()` command doesn't require any information between its parentheses.
- 6 Choose Window > Code Snippets, or if your Actions panel is already open, click the Code Snippets button at the top right of the Actions panel to open the Code Snippets panel. The code snippets are organized in folders that describe their function. In the Code Snippets panel, expand the folder called Timeline Navigation and select Click to Go To Frame and Stop. Click the Show code option to reveal the actual ActionScript code. Click the blue code instance_name_here and drag the pick whip to point to the button. Customize the code by providing the correct destination frame in the `gotoAndStop()` command. Click Insert to add the code to the Timeline.

Lesson 7: Using Text

In Lesson 7, students work with the sophisticated layout options available in the Text Layout Framework text tool to create an interactive site for a real estate company featuring information on a property and its neighborhood. As they work through the project, students learn the difference between Read Only, Selectable, and Editable TLF Text. They also learn that TLF Text depends on an external ActionScript library (a SWZ file) that is generated when their Flash movie is published or tested. Students create text with multiple columns, text that wraps around objects on the Stage, and hyperlinked text, and use ActionScript to read user-inputted text and dynamically change text on the Stage.

Understanding TLF Text

Since the text tool comes in two flavors, you should explain how TLF Text differs from Classic Text, which is still an option for targeting previous versions of the Flash Player, or for creating text that doesn't require sophisticated layouts. Students will likely wonder about the extra file (SWZ) that is generated when they publish or test their movie. TLF Text depends on a specific external ActionScript library to function properly. The SWZ file is the external ActionScript library that supports TLF Text. When a SWF file that contains TLF Text is playing from the Web, the SWF looks for the library in a couple of different locations. The SWF looks for the library on the local computer it is playing on, where the library is usually cached from normal Internet usage. The SWF also looks on Adobe.com for the library file, and if that fails, looks in the same directory as the SWF. Students should always keep the SWZ file with their SWF file so the TLF Text features work properly when they test their movies locally.

Adding text

Adding text is a rather straightforward process. Encourage students to explore many of the formatting options available in the Properties inspector. You should also remind students that text remains editable, which means that they can always double-click the text box to edit the text or change its formatting options. Students can break apart text (Modify > Break Apart) to convert each letter into separate drawing objects whose stroke and fill they can modify. However, once broken apart, the text will no longer be editable.

Adding multiple columns

In the Container and Flow section of the Properties inspector, you can change a text box to have the text flow in multiple columns. As text in the first column reaches the bottom of the text box, the next line appears at the top of the next column. In the Container and Flow section of the Properties inspector, students can

also modify the gutter, the space between columns, and the padding between the text and the boundaries of the text box to help get the exact text layout they desire.

Wrapping text

You create wrapping text by linking separate text boxes. Text will flow from one text box to another as if in a single container. It's useful to show students multiple examples of wrapping text and why it makes a more pleasing, integrated design. You could pull examples from Web sites or from print magazines.

To create linked text boxes, click and drag out the first text box on the Stage. Click on the empty box on the lower-right corner of your text box. Your cursor changes to an icon of the corner of a text box. Click and drag out your second text box. When you release your mouse cursor, your second text box becomes linked to the first text box. Blue lines that connect the first to the second represent the linkage. Explain to students how you can resize individual text boxes, break the links between text boxes, or reestablish the links.

Creating user-input text

Students create a Mortgage Calculator, which accepts input from the user through the keyboard, and displays estimated monthly payments based on those inputs. Students create user-input text with Editable TLF Text. For students who may be familiar with the older way of using text, they may still distinguish between “dynamic” text and “static” text. However, in TLF Text, all text is essentially dynamic. Any text box created with TLF Text can be given an instance name in the Properties inspector and be controlled with ActionScript. The contents of a TLF text box are referenced by the keyword `text`. Although the Mortgage Calculator part of this project may appear complicated, emphasize for students two take-home messages: First, the contents of text boxes are referenced by the text property. Second, text boxes contain String data. To make numeric calculations, you must first convert the text to a number with `Number()`. To convert a number back to text, you use `String()`.

Embedding fonts and device fonts

For any text that may be edited during runtime, you should remind students to embed their fonts. Since the user can enter any kind of text in editable text boxes, you need to include those potential characters in the final SWF to ensure that text appears as you expect it, with the same font that you've chosen in the Properties inspector. The alternative to embedding fonts is to use device fonts. These options find and use the fonts on a user's computer that most closely resemble the specified device font. When you use device fonts, you don't have to worry about embedding fonts, and you can be assured that your viewer sees text that is similar to the text that you see in the authoring environment.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

- 1 How can you tell if there is too much text to fit inside a text box, and how can you fix it?
- 2 How do you add a hyperlink to text?
- 3 What is the Target option in hyperlinked text?
- 4 Why would you need to use the ActionScript commands `String()` or `Number()`?
- 5 What are device fonts?

Answers

- 1 If your text doesn't entirely fit in its text box, Flash displays a red cross at the bottom-right corner. The red cross means there is overflow text that is not visible. To see more of the text, you can enlarge the text box. With the Text tool or the Selection tool, click and drag the edges of the text box to make it wider or taller. The text box is resized, and the text reflows to accommodate the new dimensions. You can also resize a text box by entering new Width and Height values in the Properties inspector.
- 2 Select the words you want to be hyperlinked. In the Advanced Character section of the Properties inspector, enter the URL for the link and choose `_blank` in the Target pull-down menu. The selected words in your text box become underlined, indicating that it is hyperlinked.
- 3 The Target field determines where the Web site loads. A `_blank` Target means that the Web site loads in a blank browser window. A `_self` Target loads the URL in the same browser window, taking over your Flash movie. A `_top` and `_parent` Target refers to the arrangement of a frameset and loads the URL in a particular frame relative to the current frame.
- 4 Text boxes contain text, or `String` data. To convert the contents of a text box into a number for the sake of calculations, use `Number()`. To convert a number back to text, you use `String()`.
- 5 Device fonts are three generic options grouped at the top of the Character Family pull-down menu. Use device fonts as an alternative to embedding fonts. You can also choose the Use device fonts option from the Anti-alias pull-down menu. The three device fonts are `_sans`, `_serif`, and `_typewriter`. These options find and use the fonts on a user's computer that most closely resemble the specified device font. When you use device fonts, you don't have to worry about embedding fonts, and you can be assured that your viewer sees text that is similar to the text that you see in the authoring environment.

Lesson 8: Working with Sound and Video

In Lesson 8, students bring audio and video files together to create a dynamic, interactive wildlife preserve kiosk. As they work through the project, students learn how to edit audio files, use Adobe Media Encoder CS6 to encode their video files to the correct format, and integrate video into their Flash project. Students also learn to use cue points to have their video trigger certain events so they can synchronize Flash elements to the video.

Using sounds

To add an audio file to a frame in Flash, first import it to the Library panel. Then select the frame at which you want the audio file to begin playing. In the Properties inspector, choose the audio file from the Sound menu. You can also simply drag the sound file from the Library panel to the Stage.

Selecting the appropriate option from the Sync menu is important. If a sound file doesn't play when you think it should, check the Sync menu to ensure you've selected the right option:

- **Event** synchronizes the sound to the occurrence of an event, such as a button click. An event sound plays when its first keyframe appears and plays in its entirety, even if the SWF file stops playing. If another event occurs, the first instance of the sound continues playing while the second instance of the sound plays.
- **Start** synchronizes the sound to the occurrence of an event, just as the Event option. However, if the sound is already playing, no new instance of the sound plays.
- **Stop** silences the specified sound.
- **Stream** synchronizes the sound for playing on a Web site. Flash forces animation to keep pace with stream sounds. If it can't draw animation frames quickly enough to keep up with the sound file, it skips frames. Stream sounds stop if the SWF file stops playing. Next to the Sync menu is a menu that determines how many times the sound file plays.
- **Repeat** specifies the number of times the file plays; specify a number.
- **Loop** repeats the sound continuously.
- **None** applies no effect.
- **Left Channel/Right Channel** plays sound in the left or right channel only.
- **Fade Left To Right/ Fade Right To Left** shifts the sound from one channel to the other.
- **Fade In** gradually increases the volume of a sound over its duration.
- **Fade Out** gradually decreases the volume of a sound over its duration.
- **Custom** lets you create custom in and out points of sound using Edit Envelope

● **Note:** Repeating stream sounds is not recommended, because frames are added to the file and the file size is increased by the number of times the sound is repeated. You can also apply effects to the sound file.

Editing a sound clip

In Flash, you can change the point at which a sound starts and stops playing, or control the volume of the sound as it plays. In this lesson, students clip the end of the waveform of a sound to shorten the sound clip and edit the volume levels:

- To edit a sound file, first add it to a frame, and click Edit in the Properties inspector.
- To change the start and end points of a sound, drag the Time In and Time Out controls in the Edit Envelope.
- To change the sound envelope, drag the envelope handles to change levels at different points in the sound. Envelope lines show the volume of the sound as it plays. To create additional envelope handles (up to eight total), click the envelope lines. To remove an envelope handle, drag it out of the window.
- To display more or less of the sound in the window, click the Zoom In or Out button.
- To switch the time units between seconds and frames, click the Seconds and Frames buttons.

Understanding Flash Video

Before you start working with video, consider what video quality you need, what video format to use, what size can accommodate your layout, and how it will be integrated with the rest of your Flash project. The two ways in which you can use video in Flash is (1) to embed the video, or (2) to play the video from an external file. Embedding video requires that the video be in FLV format; playing from an external file requires that the video be in FLV or F4V format.

When you embed a video, it increases the size of the final SWF file that you publish. The embedded video appears on the Timeline and can be synchronized with other Flash elements.

When you play external video, the final SWF remains small because the video is kept separate from the Flash document. Playback of external video depends on a video playback component, which you can configure, and an optional skin, which provides the look and feel and functionality of the playback controls.

Using Adobe Media Encoder CS6

You can convert your video files into the FLV or F4V format using Adobe Media Encoder CS6, a stand-alone application that comes with Flash CS6. Adobe Media Encoder can convert single files or multiple files (known as batch processing) to make your workflow easier.

You can customize many settings in the conversion of your original video to the Flash Video format. You can crop and resize your video to specific dimensions, just convert a snippet of the video, adjust the type of compression and the compression levels, or even apply filters to the video. To display the encoding options, click the Preset selection in the display window, or choose Edit > Export Settings. The Export Settings dialog box appears.

To simplify the conversion process, you can rely on the preset settings, which offer common encoding profiles for a one step operation.

Playback of external video

To play back an external FLV or F4V file, in Flash choose File > Import > Import Video. The Import Video wizard appears, which guides you through the process. During the process, you choose the external file that you want to play and an optional skin, which provides the look and feel and functionality of the controls.

Using external video files provides the following capabilities that are not available when using embedded video:

- **Long video clips.** You can use longer video clips without slowing down playback. External video files are played using cached memory, which means that large files are stored in small pieces and accessed dynamically; they do not require as much memory as embedded video files.
- **Different frame rate.** An external video file can have a different frame rate from the Flash document in which it plays. For example, you can set the Flash document frame rate to 30 fps and the video frame rate to 21 fps, which gives you greater control in ensuring smooth video playback.
- **No interruption.** With external video files, Flash document playback does not have to be interrupted while the video file is loading. Imported video files can sometimes interrupt document playback to perform certain functions (for example, to access a CD-ROM drive). Video files can perform functions independently of the Flash document, so they do not interrupt playback.
- **Captions and cue points.** Captioning video content is easier with external video files because you can use callback functions to access metadata for the video. Cue points are special markers that can be tied to ActionScript code for additional functionality.

Embedding Flash Video

Embedded video lets you insert a video file directly within a SWF file. When you import video in this way, the video is placed in the Timeline where you can see the individual video frames represented in the Timeline frames. An embedded video file becomes part of the Flash document.

When you create a SWF file with embedded video, the frame rate of the video clip and the SWF file must be the same. If you use different frame rates for the SWF file and the embedded video clip, playback is inconsistent.

Embedded video works best for smaller video clips with a playback time of less than 120 seconds. If you are using video clips with longer playback times, consider keeping and playing the video as an external file.

The limitations of embedded video include:

- **Performance problems.** You might encounter problems if the resulting SWF files become excessively large. Flash Player reserves a lot of memory when downloading and attempting to play large SWF files with embedded video, which can cause Flash Player to fail.
- **Synchronization issues.** Longer video files (over 120 seconds long) often have synchronization issues between the video and audio portions of a video clip. Over time, the audio track begins playing out of sequence with the video, causing a less than desirable viewing experience.
- **Maximum length.** There is a maximum length of 16,000 frames for embedded movies.

Despite these limitations, embedded video has a place for small, well-placed video segments.

Using cue points

Cue points are special markers that you place in your video that Flash can detect with ActionScript. There are two ways to use cue points. Cue points can trigger an ActionScript command, letting you synchronize the video with other Flash elements. Or, you can jump to particular cue points in the video with ActionScript. Both types of cue points add more functionality to video. There are three ways to add cue points to video. First, you can add cue points to video with Adobe Media Encoder CS6. Second, you can add cue points to a selected video on the Stage in the Properties inspector. Third, you can add cue points with ActionScript, which was not discussed in this lesson.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

- 1 How do you set the quality of sounds?
- 2 How do you fade out the volume of a sound?
- 3 What are the Sync options for sound?

- 4 Which video formats are compatible with Flash?
- 5 Which of these options are *not* available in Adobe Media Encoder CS6? (1) Cropping a video, (2) Trimming the length of a video, (3) Combining multiple video clips together, (4) Stripping the audio from a video, (5) Changing the frame rate of a video, (6) Adding cue points to video.
- 6 What is the SkinAutoHide option, and how do you enable it?
- 7 If your video has a transparent background, what is the required format for your video?

Answers

- 1 Set the sound quality and compression in the Publish Settings options. Select the Flash format from the left and click on the Audio Stream and Audio Event values. In the Sound Settings dialog box that appears, change the bit rate, which determines the quality of the sound in your final, exported Flash movie. The higher the bit rate, the better the quality.
- 2 You can change the volume levels in the Edit Envelope dialog box. Select the first keyframe that contains your sound and in the Properties inspector, click the Pencil button. The Edit Envelope dialog box appears. Click on the top horizontal line of the waveform to add keyframes. Drag the keyframes down to the bottom of the window to reduce the volume at that point in time. You can also select the Fade out option from the Effect menu in the Properties inspector to apply a global fade from start to finish.
- 3 Sound sync refers to the way the sound is triggered and played. There are several options: Event, Start, Stop, and Stream. Stream ties the sound to the Timeline so you can easily synchronize animated elements to the sound. Event and Start are used to trigger a sound when the playhead enters the keyframe. Event and Start are similar except that the Start sync does not trigger the sound if it is already playing (so there are no overlapping sounds possible with Start sync). The Stop option is used to stop a sound.
- 4 The video formats that are compatible with Flash are FLV, F4V, and video encoded with the H.264 codec.
- 5 Option 3: You cannot use Adobe Media Encoder CS6 to combine multiple video clips together.
- 6 To hide the video controller and only display it when users roll their mouse cursor over the video, use the SkinAutoHide option. Select your video on the Stage, and enable the SkinAutoHide option in the Component Parameters section of the Properties inspector.
- 7 If you want to preserve a transparent background, your video must be encoded as an FLV with the On2 VP6 codec.

Lesson 9: Loading and Controlling Flash Content

In this lesson, students learn how to keep a Flash project modular by loading external SWF files into a main SWF. In addition, they learn how to control a movie clip's Timeline and learn to create masks. The lesson is based on a fictional life-style magazine. Each section of the magazine is a separate SWF file that loads into the main SWF.

Loading external content

Loading external content keeps your overall project in separate modules and prevents the project from becoming too bloated and difficult to download. It also makes it easier for you to edit, because you can edit individual sections instead of one, large, unwieldy file.

Loading a SWF involves ActionScript. You can either write the code in the Actions panel yourself or use the Code Snippets panel to help. A ProLoader object must first be created, as in the following statement:

```
var myProLoader:ProLoader=new ProLoader();
```

Next, the load() method should be called for the ProLoader object. The parameter for the load() method is the URLRequest object, which defines the external SWF file.

```
var myURL:URLRequest=new URLRequest("page1.swf");  
myProLoader.load(myURL);
```

Be sure to point out to students that the filename for the SWF file must be enclosed in double quotation marks.

Then, to display the loaded SWF on the main Stage, you need to add the ProLoader object to the display list with the following command:

```
addChild(myProLoader);
```

In this lesson, students enter the code inside four separate event handlers.

Explain to students that there are two different ways to remove the loaded content. One way is to remove the ProLoader object from the Stage with removeChild(). The second way is to unload the loaded content with unload(). Make sure they know the difference between these two approaches.

Controlling movie clips

The initial animations are nested inside individual movie clips, and students learn to control each of those movie clips' Timelines. After any one of the external SWFs is removed from the Stage, the playhead moves to the first frame of each of the movie clips and plays. Students learn to use the basic navigation commands that they learned in Lesson 6 (gotoAndStop, gotoAndPlay, stop, play) to navigate the Timelines of movie clips.

● **Note:** For those familiar with ActionScript, the ProLoader object is the newer version of the Loader object. The methods and application of the ProLoader is identical to the Loader object, but the ProLoader object handles the loading of SWF files with TLF text better.

To navigate the Timeline of a movie clip, simply precede the navigation command with the name of the movie clip and separate them with a dot. Flash targets that particular movie clip and moves its Timeline accordingly.

Creating masks

Masking is a way of selectively hiding and displaying content on a layer. Masking allows you to control the content that your audience sees. Provide several examples of the kinds of effects that masking can do: For example, you can make a circular mask to create a spotlight effect, and you can animate both the shape of the spotlight or the image under the spotlight.

In Flash, you put a mask on one layer and the content that is masked in a layer below it. After the students grasp the basic concept of masks, reveal the possibilities of an animation in the Mask layer or an animation in the Masked layer.

To create a mask, double-click the Layer icon in front of the layer name. In the Layer Properties dialog box that appears, select Mask. That layer becomes a Mask layer. Anything that is drawn in this layer will act as a mask for a masked layer below it.

To create the Masked layer, simply drag another normal layer under the Mask layer. Or double-click the Layer icon in front of the layer name in the layer below the Mask layer. In the Layer Properties dialog box that appears, select Masked.

Warn students of the limitations of masking: Flash does not recognize different Alpha levels of a mask. For example, a mask at an Alpha value of 50% will still mask at 100%. However, with ActionScript you can dynamically create masks that will allow transparencies. Masks also do not recognize strokes.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

- 1 What is the difference between using `removeChild()` and `unload()` to remove the loaded content?
- 2 How do you position the `ProLoader` object on the Stage?
- 3 What do you see in a Masked layer when a shape in the Mask layer is filled with a color at 50% opacity?

Answers

- 1 If you use `removeChild()`, the `ProLoader` object (along with its content) on the Stage is removed from the display. If you use `unload()`, the `ProLoader` object remains on the Stage, but any content that has been loaded into it is removed.
- 2 You can position the `ProLoader` object by assigning a new value to its `x` or `y` properties with `ActionScript`. In the Actions panel, enter the name of the `ProLoader` object, followed by a period, the property `x` or `y`, an equals symbol, and a new value.
- 3 Opacity in the Mask layer has no effect on what you see in the Masked layer.

Lesson 10: Publishing Flash Documents

In Lesson 10, students explore a range of options for testing and publishing a Flash project. They publish a finished banner animation for the Web, they create a desktop application with Adobe AIR, and they test a mobile app within the Flash application.

Understanding publishing

Publishing is the process that creates the required files to play your final Flash project for your viewers. Students need to understand the difference between *author-time* and *runtime*. Authortime refers to development within the Flash CS6 application since it is the environment in which content is being created, or authored. Runtime refers to playback in the targeted viewing environment, such as a desktop browser or a mobile device. Adobe provides various runtime environments to play Flash content. The most common is the Flash Player, which plays Flash content on Web browsers. Adobe AIR is another runtime environment that can play Flash content directly from your desktop, or can play Flash content as an app on mobile devices running Android or the iOS.

Testing a Flash document

One fast way to preview a movie that targets the Flash Player is to choose `Control > Test Movie >` in Flash Professional (`Ctrl+Enter/Command+Return`). This command creates a SWF file in the same location as your FLA file so you can play and preview the movie; it does not create the HTML file necessary to play the movie from a Web site. Remind students that the default behavior for the movie in the Test Movie mode is to loop. You can make your SWF play differently in a browser by selecting different publish settings, as described later in this lesson, or by adding `ActionScript` to stop the Timeline. Another option is to choose `File > Publish Preview > Default (HTML)` to export a SWF file and an HTML file required to play in a browser and to preview the movie.

Publishing a movie for the Web

When you target the Flash Player, the Publish command creates a Flash SWF file and an HTML document that inserts your Flash content in a browser window. If you change publish settings, Flash saves the changes with the document. The default settings for Publishing should be sufficient for most students, but it's useful to point out the various options available for customization.

Understanding the bandwidth profiler

You can preview how your final project might behave under different online conditions by using the Bandwidth Profiler, a useful panel that is available when you are in Test Movie mode.

When simulating the download speed, Flash uses estimates of typical Internet performance, not the exact modem speed. For example, if you select to simulate a modem speed of 28.8 Kbps, Flash sets the actual rate to 2.3 Kbps to reflect typical Internet performance. The profiler also compensates for the added compression support for SWF files, which reduces the file size and improves streaming performance.

To simulate download performance:

- 1 Choose Control > Test Movie > in Flash Professional.
- 2 In the preview window, choose View > Download Settings.
- 3 Select a download speed to determine the streaming rate that Flash simulates. When viewing the SWF file, choose View > Bandwidth Profiler to see a downloading performance graph.
 - The left side of the profiler displays information about the document, its settings, its state, and streams, if any are included in the document.
 - The right section of the profiler shows the Timeline header and graph. In the graph, each bar represents an individual frame of the document. The size of the bar corresponds to that frame's size in bytes. The red line beneath the Timeline header indicates whether a given frame streams in real time with the current modem speed set in the Control menu. If a bar extends above the red line, the document must wait for that frame to load.
- 4 Choose View > Simulate Download to turn streaming off or on. If you turn on streaming, the document starts over and simulates a Web connection.

Publishing an AIR application

You can create a desktop application for Windows or for the Macintosh by targeting Adobe AIR 3.2 for the Desktop in the Publish section of the Properties inspector. An AIR application requires the AIR runtime. You can either download the free AIR runtime from Adobe's site, or you can include the AIR runtime within your application during the publishing process.

To create an AIR desktop application, edit the AIR application settings and provide some additional information:

- You need to provide a certificate so that users can trust and identify the developer of the Flash content. In the Signature section of the Application Settings dialog box, you can guide your students to create their own self-signed certificate, which has the .p12 extension.
- The Icons section of the Applications Settings dialog box allow students to use PNG images as icons for the AIR application.
- The Advanced section of the Applications Setting dialog box contains settings for the initial window location and options for resizing.

Simulating mobile app interactions

Flash CS6 has a new feature called the SimController that allows you to simulate mobile app interactions. Since creating an app for mobile devices is a little more complicated because it requires specific developer certificates and the actual hardware, the SimController is useful for early testing. Interactions such as swiping and pinching, or even tilting the device, can be emulated directly within Flash CS6.

In this lesson, students use the SimController to test mobile interactions in an already completed Flash file that targets AIR 3.2 for Android. To test a mobile app, choose Control > Test Movie > in AIR Debug Launcher (mobile), and the SimController automatically launches.

Mobile interactions that can be tested in the SimController include:

- Using the accelerometer in three directions (x-, y-, and z-axes)
- Click and drag, press and tap, two-finger tap, pan, rotate, zoom, and swipe
- Geolocation
- Menu, Back, and Search buttons

Optimizing Flash documents

As your document file size increases, so does its download time and playback speed. You can take several steps to prepare your document for optimal playback. As part of the publishing process, Flash automatically performs some optimization on documents. Before exporting a document, you can optimize it further by using various strategies to reduce the file size. Students should be made aware of the importance of file optimizations, because even the most gorgeous and astounding animations will never be seen if they are too large and cumbersome to download quickly. As you make changes, test your document by running it on a variety of computers, operating systems, and Internet connections.

Follow these guidelines to best optimize your files:

- Use symbols, animated or otherwise, for every element that appears more than once.
- Use tweened animations whenever possible when creating animation sequences. Tweened animations use less file space than a series of keyframes.
- Limit the area of change in each keyframe; make the action take place in as small an area as possible.
- Avoid animating bitmap elements; use bitmap images as background or static elements.
- Use MP3, the smallest sound format, whenever possible.
- Group elements.
- Use layers to separate elements that change during the animation from elements that do not.
- Use Modify > Shape > Optimize to minimize the number of separate lines that are used to describe shapes.
- Limit the number of special line types, such as dashed, dotted, ragged, and so on. Solid lines require less memory. Lines created with the Pencil tool require less memory than brushstrokes.
- Limit the number of fonts and font styles. Use embedded fonts sparingly because they increase file size.
- For Font Embedding options, select only the characters needed instead of including the entire font.
- Use the Color Effect menu in the Properties inspector to create different colored instances of a single symbol.
- Use gradients sparingly. Filling an area with gradient color requires about 50 bytes more than filling it with solid color.
- Use alpha transparency sparingly because it can slow playback.
- Use the high quality setting on filters sparingly because it can slow playback.

Bitmap caching

Bitmap caching helps you enhance the performance of nonchanging movie clips in your applications. When you set the Render mode for a movie clip instance to Cache as Bitmap, Flash Player caches an internal bitmap representation the instance. This can improve performance for movie clips that contain complex vector content. All the vector data for a movie clip that has a cached bitmap is drawn to the bitmap instead of to the main Stage.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

- 1 What is the purpose of QA testing?
- 2 How do you simulate how your Flash movie performs with a particular Internet connection?
- 3 Why do you have to merge code when you create a projector file of a Flash movie that contains TLF Text?
- 4 What kind of file has a .p12 extension, and why is it required?

Answers

- 1 QA, or quality assurance, testing is the process where the functionality of a project is tested. QA testing is important to measure the actual performance with the expected performance of any multimedia project.
- 2 You can simulate different download speeds in Test Movie mode with the Bandwidth profiler. Choose View > Download Settings, and then choose a particular type of Internet connection. To see how the movie performs under those conditions, choose View > Simulate Download.
- 3 Text Layout Format Text depends on an external ActionScript library to function correctly. To create a projector that can stand alone, you need to merge the external ActionScript library with the projector file.
- 4 A .p12 file is a certificate that is required to publish an Adobe AIR application. It identifies the developer of the Flash content so users can trust to install the application on their desktop. If you don't have an official certificate, you can create a self-signed certificate within Flash CS6.